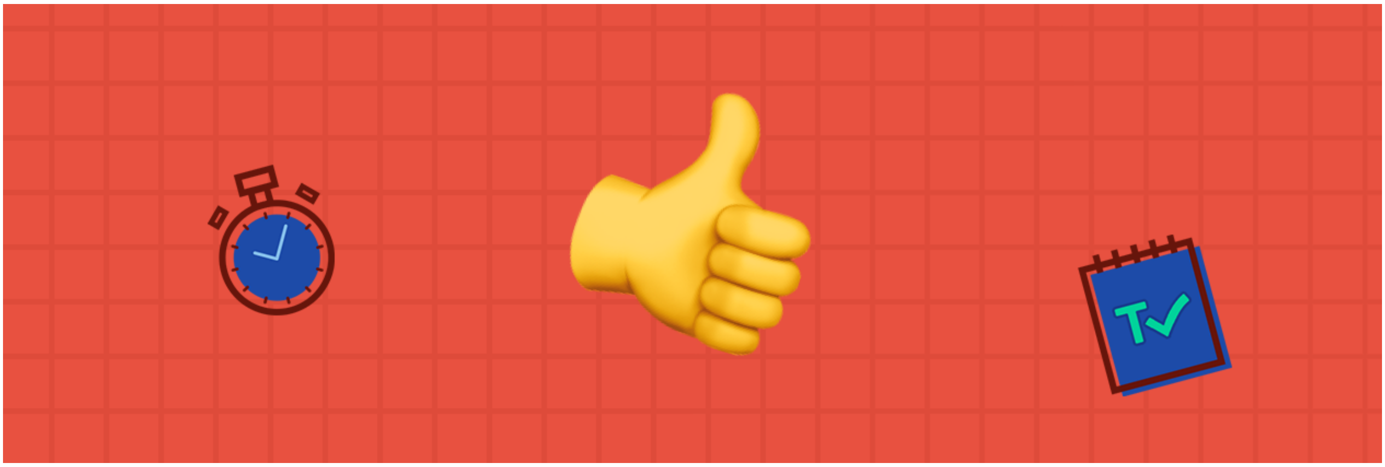
【03】推荐序 | 最合适的作者和最特殊的工具



2017 年 7 月的第一个工作日开始，有一个头像特别突出的作者在 Matrix 隔 1 天就投 1 篇文章，而且每一篇文章的切入点都不一样。当这个人发到第四篇稿时，我意识到，他在这个圈里是个不应该被埋没的新星。

随后他笔耕不赘，Workflow、Automator、树莓派等进阶工具皆有涉猎，同时也可以写非常方法论的，人人都能理解和领会的文章。

成为签约作者之后，因为速度和质量，他很快就成了编辑部约稿的第一人选、作者群中公认的「稿霸」。甚至一度编辑部认为分配给他的任务太多了，已经不好意思给他再分稿的时候，他却在群里说现在无稿可写，又自己折腾出一个新玩意在写了。

这个作者就是 Minja，少数派季度最佳作者，现在已经正式成为了少数派编辑部的一员。

要做到 Minja 写稿这个速度和质量，首先需要的就是任务管理能力。因为在少数派作者身份之外，他也有自己的事要做。一般来说，执行力这么高的人，不会只有一个方面执行力高。那么，如果他任务理不清、优先度排不好，他就算每天只写稿，其他的事务最终也会把他吞噬，高效不会持续。

其次他需要有极强的执行力，才能将他的任务按计划进行。写稿这种任务不是必须做的，想拖随时可以拖，但是他不仅不拖而且高产非常。能够做到这点我个人是很服气的，任务管理我是没问题的，但一天让我写三篇大文章，我肯定不乐意。

除此之外，作为一个如此高产的约稿作者，他在效率工具方面固然有足够的基础，但同样，他也要可以迅速对复杂的软件功能攻坚，还要能在最短时间内把攻坚的结果转化成人们能够理解的文字，再找到人们有共感的例子，最后将其组织成初学者读得明白的文章。

说完了 Minja，再来说说 Taskpaper，这个软件的由来非常有意思。

2001 年到 2007 年，@jessegrosjean 想要开发一款名为 Mori 信息管理工具（Information Manager）。开发过程中他添加了他能想到的所有功能，但是他发现自己从来没有用过这个工具。反而，在开发的过程中，他把所有的 to-do 都写在了文本编辑器里。起初他不太甘心，想要尝试使用 Mori，但总是会回到文本编辑器。所以，最后他卖掉了 Mori，投入到了基于纯文本的任务管理工具的开发。

纯文本的优势是还原了我们对记录任务时的体验，这让我们在添加任务时非常舒服：按一下 Return 换行就是一个新任务，想加标签就直接加 @……在纸上记任务不就是这种体验吗?

当然，仅仅用纯文本是不够的，任务管理不仅是记录，也要整理、过滤。所以 Taskpaper 以不破坏笔记本的简单纯粹为前提，加入了高度灵活和进阶的过滤功能和语法，使其能够满足复杂的任务系统。

就我所知，Overcast 的作者 Marco Arment、Editorial 的作者 Ole Zorn、Marked 的作者 Brett Terpstra 这些 Geek 都在使用 Taskpaper，足以说明它可以胜任复杂的独立开发任务。读过我教程或者听过我播客的人大概知道，我现在对任务管理工具的选择强调「现代」这两个字，而这两个字主要体现在移动设备上的表现和云属性。但大家都不知道，在 16 年初 Taskpaper 3 公测时，我就想要给它写一篇长篇测评。当时文章的结构已经确定，基本资料也收集完毕，草稿也已经写了三四千字。

这篇文章终究搁置的原因是其时我的工作环境已经几乎主要在 iOS 上了，而 Taskpaper 是一个主打桌面系统的工具，这让我没有解决使用 Taskpaper 时遇到的困难的动力。因此我始终觉得对 Taskpaper 有所亏欠，它是个非常好的工具，却由于我的原因迟迟没有正式出现在少数派读者的视野之中。

所以，当 Minja 来找我说要写 Taskpaper 的付费教程时，我不禁额手称庆。监制教程时自然也是尽我所能，提供了所有

之前收集的资料以及草稿。在我审核时，也经常要他从头到尾结构调换甚至重写，有的还不止重写了一遍。

在这个教程里，Minja 写下了他是如何通过 Taskpaper 做到「今日事今日毕」的，有他自己的方法论、独特的技巧和巨细靡遗的功能解说。

推荐大家阅读。